Using a RAM-based sorting processor

An alternative to using a CAM of any type can be done by having a fast linked list sorting processor. By utilizing fast memory, spans could be sorted into a linear list in the same order as described above (using an SMCCAM). But, since spans are added and deleted frequently, this approach might be every slow. The advantage is the ability to use fast off-the-shift RAMs that are readily available.

To aid in SOT Query operations, a two-dimensional data structure could be used that sorts the spans in both the x and z dimensions at the same time. Such a searching and sorting structures is described on pages 24 to 55 of "Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry", by Kurt Mahlhorn, a volume in the EATCS series on Monographs on Theoretical Computer Science, edited by Brauer, W., et al., ISBM 0-387-13642.8, published by Springer-Verlag, N.Y., 1985.

Shadow processing

Because shadow computations as essentially the same as hidden surface removal, the Span Sorting Renderer **500** can be used for shadow computation. This can be done by utilizing multiple Span Sorting Renderers **500** in parallel, or it can be done by time-sharing one Span Sorting Renderer **500**.

What is claimed is:

1. A sorting magnitude comparison content addressable memory (SMCCAM) apparatus comprising:

a plurality of addressable memory storage bits, each said storage bit for storing a data bit, said memory storage bits arranged into a plurality of words;

an input circuit providing an input comprising a plurality of input bits matching some of said data bits so as to have a one-to-one bit correspondence to said data bits;

a comparator circuit simultaneously comparing said plurality of input bits to data bits in all said words, said comparator circuit making simultaneous comparisons such that each said data bit is compared to its corresponding input bit, and said comparator circuit generating a query result for each said word which query results has a first state when all said data bits within said word which are compared to one of said input bits compare favorably to each corresponding input bit, and a second state when said bits do not compare favorably;

a flag memory storage storing a flag bit generated from said query result for each of said words;

an up-counter circuit for each said word, said up-counter circuit being conditionally incremented according to said flag bit, the up-counter used to indicate a position in an ordered list; and

a circuit locating the smallest value in said up-counter circuits.

2. In a graphical processing system for processing 3-dimensional object geometry data and rendering at least some of said object geometry data on a 2-dimensional display screen, said object geometry comprising polygons, a method for rendering a display raster line comprising the steps:

(i) maintaining a list of current polygons that intersect a current display raster line;

(ii) generating a polygon span for each polygon that intersects said current display raster line based on geometric properties of said polygon including said polygon parameters, each said polygon span including subraster information describing the geometric shape of said span within a vertical extent of said display raster line;

(iii) generating at least one dummy span, said dummy span representing a location and a clipping function of a clipping plane, each said dummy span including subraster information describing the geometric shape of said span within a vertical extent of said display raster line;

(iii) storing said geometric properties of each said generated span into a span memory, said generated spans, said current span portion comprised of a set of current subspans, each said

(iv) maintaining a current span portion that is part of a potentially visible one of said generated spans, said current span portion comprised of a set of current subspans, each said current subspan representing a rectangular area within said current span portion, and said set of current subspans approximating an area of said current span portion;

(v) performing at least one span occluding test to find any new span that potentially occludes said current span portion, where said span occluding test comprises:

(1) determining the leftmost, rightmost, and farthest spatial coordinates in said set of current subspans; and

(2) performing a query operation on the said stored geometric properties in said span memory to find all said spans whose stored geometric properties include a spatial coordinate located between said leftmost and said rightmost spatial coordinates of the said set of current subspans, and a spatial coordinate closer than said farthest spatial coordinate of the said set of current subspans;

(vi) generating a set of new subspans, each said new subspan representing a rectangular area within said new span, and said set of new subspans approximating an area of said new span;

(vii) for each said subspan in said set of current subspans, performing a subspan comparison comprising:

(1) performing a spatial comparison between said subspan in said set of current subspans and a corresponding subspan in the said set of new subspans; and

(2) determining the visibility, partial visibility, or non-visibility of each subspan in said set of current subspans, taking into account the clipping function of one of the dummy spans if the new span is a dummy span; and

(viii) updating said current span portion based on result of said subspan comparisons.

* * * * *